PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06F 9/34 — A1

(11) International Publication Number: WO 00/22513

(43) International Publication Date: 20 April 2000 (20.04.00)

(21) International Application Number: PCT/GB99/03167

(22) International Filing Date: 11 October 1999 (11.10.99)

(30) Priority Data:

9822074.2	10 October 1998 (10.10.98)	GB
60/115,954	14 January 1999 (14.01.99)	US

(71) Applicant *(for all designated States except US)*: THE VICTORIA UNIVERSITY OF MANCHESTER [GB/GB]; Oxford Road, Manchester M13 9PL (GB).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: SANDHAM, John [GB/GB]; The Victoria University of Manchester, Oxford Road, Manchester M13 9PL (GB).

(74) Agent: HOLMES, Matthew, Peter; Marks & Clerk, Sussex House, 83–85 Mosley Street, Manchester M2 3LG (GB).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published

With international search report.

(54) Title: ENDIAN TRANSFORMATION

(57) Abstract

A method for emulating a processor of a first endian type on a processor of a second endian type, wherein each memory access address B of string length L is transformed to the address A–B–L+S, wherein A is the total number of bytes allocated to a program, and S is the start address of the program.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

Endian Transformation

This invention relates to an endian transformation method and system.

A problem commonly encountered by emulation systems, which run identical software on different computer processor chips is format incompatibility. One aspect of this incompatibility resides in the format in which strings of data (eg. 2-byte words or 4-byte words) are expressed. In many computer architectures, each byte of a 4-byte word has its own individual memory address; this gives rise to two possibilities for numbering the bytes within a word. In a big-endian convention, the word whose bytes are addressed (X, X+1, X+2 AND X+3) has its most significant byte addressed X, while in a little-endian convention, the address ordering is the reverse of this so that the least significant byte is addressed X and the most significant byte is addressed X+3. Other "endian formats" are known in which different conventions are observed for ordering the significance of bytes within words, but in most commercially available systems either the big-endian or little-endian convention is observed. The ordering of bits within each byte is the same whether the memory address convention is big-endian or little-endian.

Emulation systems are available which enable software (such as an operating system or an application program) of one endian format to operate on hardware of an opposite endian format. Generally, systems of this type convert each word between endian representations on a word-by-word basis. This conversion, when required frequently, introduces a significant overhead into the time required to perform a given task.

It is an object of the present invention to provide an efficient method and system to enable software of one endian format to run on hardware of a different endian format.

According to the invention there is provided a method for emulating a processor of a first type which observes a first convention for ordering the significance of bytes within words on a second type of processor which observes a second convention for ordering the significance of bytes within words, wherein memory access addresses are transformed such that bytes stored in a memory addressed by a processor of the second type as a result of an instruction in which a

byte order in accordance with the first convention is observed are distributed in a pattern which is a mirror image of the distribution pattern of the bytes which would result if the memory was addressed by a processor of the first type in response to the said instruction.

The invention also provides a method for emulating a processor of a first type which observes a first convention for ordering the significance of bytes within words on a second type of processor which observes a second convention for ordering the significance of bytes within words, the order of the second convention being the reverse of the order of the first, wherein memory access addresses are transformed such that the offset between addresses of any two bytes stored in memory is unaltered by the transformation and the relative order of the addresses of any two bytes stored in the memory is reversed by the transformation.

The invention further provides a method for emulating a processor of a first type which observes a first convention for ordering the significance of bytes within words on a second type of processor which observes a second convention for ordering the significance of bytes within words, wherein memory access addresses are transformed such that strings of bytes in the first endian format which are stored successively by the processor operating in accordance with the second endian format aggregate in the same manner as the bytes would aggregate if the processor was of the first endian format and memory access addresses were not transformed.

The invention still further provides a method for emulating a processor of a first type which observes a first convention for ordering the significance of bytes within words on a second type of processor which observes a second convention for ordering the significance of bytes within words, wherein each memory access address B of string length L is transformed to the address A-B-L+S, wherein A is the total number of bytes allocated to a program, and S is the start address of the program.

Assuming a big-endian processor and a little-endian program, the address transformations ensure that bytes aggregate in the memory in a pattern which is a mirror image of the pattern which would have resulted if the processor had been little-endian and no address transformation had been performed. The invention will operate in the same manner for a little-endian processor and a big-endian program. It is important to note that the transformation has no effect on the ordering of bits within

each byte. The result is a system which provides a considerable time saving when compared to known endian conversion methods, which convert each string of bytes between endian representations each time that string is used.

According to the invention there is provided an endian transformation system, the system comprising means for transforming an address location of a code represented in a first endian format into an address in a second endian format, the transformation comprising introducing an offset into the address, the size of the offset being determined from the difference between the address location of the code and a predefined address location.

According to a further aspect of the present invention there is further provided a process for compiling or translating a computer program code instruction using transformed address space references in the compiled or translated code especially configured for execution on a programmable machine utilizing a corresponding predetermined convention for ordering the significance of bytes within words of said address space, said process comprising:

(a) during compilation or translation of a code instruction referring to a memory address, transforming the referenced memory address with respect to a fixed block size of memory in the predetermined programmable machine so as to change the referenced address value by an amount that is fixed for a given number of bytes being accessed in each word; and

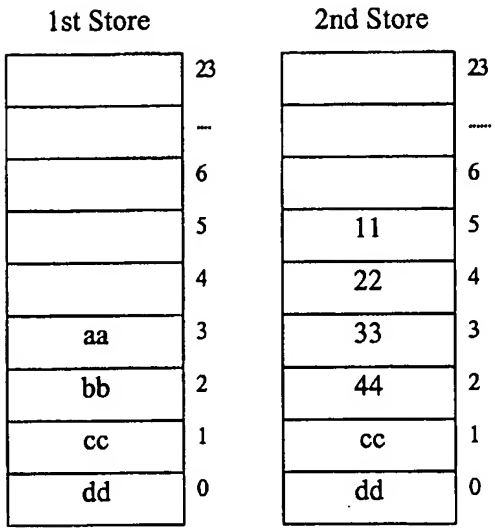
(b) including the thus changed address reference in a compiled or translated output instruction so that there is no extra operation required during execution of the output instruction to accommodate the convention for ordering bytes within words used by said predetermined programmable machine.

A specific embodiment of the invention will now be described by way of example only.

The following two assembly code store instructions

```
movl $0xaabbccdd,[0]
movl $0x11223344,[2]
```

will have the following effect in a little-endian architecture:



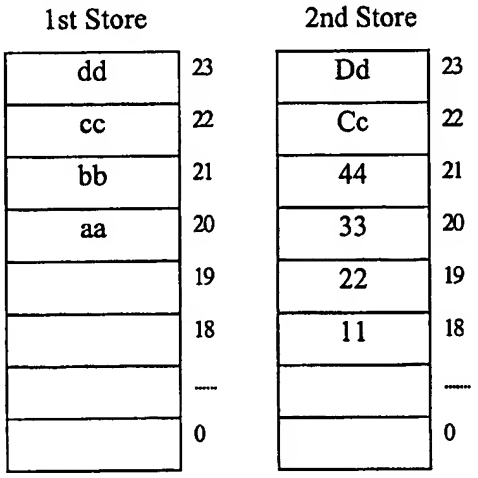
The first store instruction stores the least significant byte (dd) of the first 4-byte word (aabbccdd) at address '0', the second least significant byte at address '1', etc. The second store instruction stores the least significant byte (44) of the second 4-byte word (11223344) at address '2', the second least significant byte at address '3', etc. Thus, the effect of storing the second 4-byte word is to overwrite the prior contents of addresses '2' and '3', and the two most significant bytes of the first 4-byte word 'aa' and 'bb' are lost.

If the same two store instructions are used in a big-endian architecture, the most significant byte (aa) of the first 4-byte word will be stored at address '0', the next most significant byte (bb) at address '1', etc. The second store instruction will overwrite the prior contents of addresses '2' and '3', as before, and the two least significant bytes of the first 4-byte word 'dd' and 'cc' will be lost. Thus, when a big-endian architecture is used, the contents of an addressed memory resulting from assembly code store instructions will differ from the contents of an equivalent memory when a little-endian architecture is used.

The invention allows both big-endian and little-endian words to be stored in such a way that any given store instruction will result in the same bytes being stored in both architectures, although the order of the bytes is reversed. This is achieved by transforming memory access addresses such that the pattern of bytes stored in a memory addressed by for example a big-endian processor is a mirror image of the

pattern which would have resulted if the memory had been addressed without transformation by a little-endian processor.

In the case of the assembly code store instructions given above, in order to accommodate a big-endian architecture, the two 4-byte words are stored in accordance with the present invention at the uppermost available addresses as shown below:



To preserve information in the memory, the second of the two stores places the second 4-byte word at a lower address than the first 4-byte word, thus overwriting the same 2-byte word (0xaabb) of information as in the little-endian architecture. The effect in terms of the bytes stored in the memory of the assembly code instructions in the little-endian architecture is thus duplicated in the big-endian architecture, although the order of the bytes is reversed.

The above exemplification of the system in accordance with the present invention in relation to an unaligned store instruction demonstrates the flexibility of the system. The system may also be used for aligned store instructions.

The address transformations used as described above to preserve information in the big-endian architecture are:

access type	Adjustment
word (4-byte)	addr'=[20-addr]
word (2-byte)	addr'=[22-addr]
byte	addr'=[23-addr]

This generalises to:

access type	general adjustment	where
word (4-byte)	addr'=endianAdj_L-addr	EndianAdj_L=progSize-4
word (2-byte)	addr'=endianAdj_W-addr	EndianAdj_W=progSize-2
byte	addr'=endianAdj_B-addr	EndianAdj_B=progSize-1

Thus, using the generalisation shown in the above table, the following operations in a little-endian architecture:

movw $0xaabb,[1]

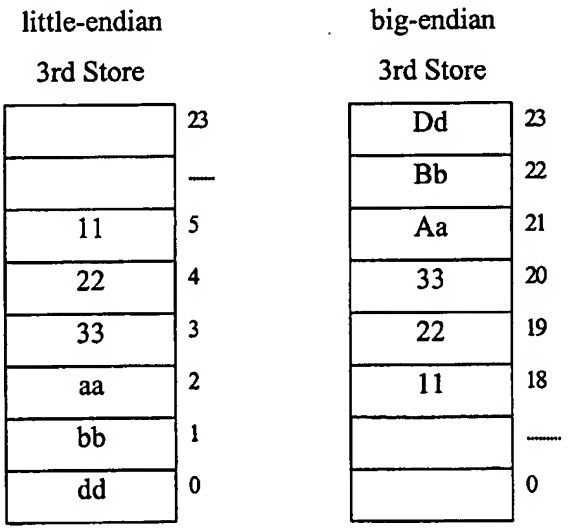
movl [1],%eax

will have the same effect as the following operations in a big-endian architecture:

movw $0xaabb,[22-1]

movl [20-1],%eax

The effect of the above commands is shown below:



The invention introduces one extra arithmetic operation for every load/store instruction. However, many instructions which access memory use address expressions which contain constant offsets such as:

addl %edx,0x8(ebp,eax,4)

which represents the effective address:

ebp+eax*4+8.

This expression, after memory access transformation in accordance with the invention has been applied, becomes:

endianAdj_L-(ebp+eax*4+8).

Folding the constants of the expression can be used to give:

(endianAdj_L-8)-(ebp+eax*4).

Thus, folding allows those terms which may be calculated at translation time to separated from those terms which are held in registers and are unknown at translation time. Since the term 'endianAdj_L' is known at translation time its effect is calculated before run time, and the memory access transformation will not cause a loss of performance at run time. Thus, in general the big-endian transformation of the invention incurs no extra overhead for the majority of memory accesses.

A subject machine program (or operating system) is treated as if it is loaded contiguously from address 0, while internally being stored as a mirror image, as shown below:

Actual Memory Configuration (big-endian)		Memory Configuration intended by the assembly code (little-endian)	
c7	23		23
45	22		.....
f8	21	00	6
03	20	00	5
00	19	00	4
00	18	03	3
00	17	f8	2
	.....	45	1
	0	c7	0

If the assembly code specifies access to the 4-byte value 0x00000003 at memory location 3, using the same program size as the previous examples, this

memory access becomes endianAdj_L-3 = (progSize-4)-3 = (24-4)-3 = 17, which is the address in the big-endian mirror image of the value required.

Whereas the above examples illustrate use of the invention in transforming code intended for a little-endian architecture so that it will run on a big-endian system, the invention could be used to transform big-endian code to run on a little-endian system. Indeed, the invention can be used to transform between any two endian systems which are byte reversals of one another.

The endian transformation method may be used as part of a complete emulation system.

The advantages joined by the "folding" operation described above are not limited to the particular transformation described. A similar operation may be performed in other compilation or translation processes using transformed address space references in the compiled or translated code to include the changed address in an output instruction to reduce overheads during execution of that output instruction.

Claims

1. A method for emulating a processor of a first type which observes a first convention for ordering the significance of bytes within words on a second type of processor which observes a second convention for ordering the significance of bytes within words, wherein memory access addresses are transformed such that bytes stored in a memory addressed by a processor of the second type as a result of an instruction in which a byte order in accordance with the first convention is observed are distributed in a pattern which is a mirror image of the distribution pattern of the bytes which would result if the memory was addressed by a processor of the first type in response to the said instruction.

2. A method for emulating a processor of a first type which observes a first convention for ordering the significance of bytes within words on a second type of processor which observes a second convention for ordering the significance of bytes within words, the order of the second convention being the reverse of the order of the first, wherein memory access addresses are transformed such that the offset between addresses of any two bytes stored in memory is unaltered by the transformation and the relative order of the addresses of any two bytes stored in the memory is reversed by the transformation.

3. A method for emulating a processor of a first type which observes a first convention for ordering the significance of bytes within words on a second type of processor which observes a second convention for ordering the significance of bytes within words, wherein memory access addresses are transformed such that strings of bytes in the first endian format which are stored successively by the processor operating in accordance with the second endian format aggregate in the same manner as the bytes would aggregate if the processor was of the first endian format and memory access addresses were not transformed.

4. A method for emulating a processor of a first type which observes a first convention for ordering the significance of bytes within words on a second type of

processor which observes a second convention for ordering the significance of bytes within words, wherein each memory access address B of string length L is transformed to the address A-B-L+S, wherein A is the total number of bytes allocated to a program, and S is the start address of the program.

5. An endian transformation system, the system comprising means for transforming a memory access address in accordance with a method according to any preceding claim.

6. A process for compiling or translating a computer program code instruction using transformed address space references in the compiled or translated code especially configured for execution on a programmable machine utilizing a corresponding predetermined convention for ordering the significance of bytes within words of said address space, said process comprising:

(a) during compilation or translation of a code instruction referring to a memory address, transforming the referenced memory address with respect to a fixed block size of memory in the predetermined programmable machine so as to change the referenced address value by an amount that is fixed for a given number of bytes being accessed in each word; and

(b) including the thus changed address reference in a compiled or translated output instruction so that there is no extra operation required during execution of the output instruction to accommodate the convention for ordering bytes within words used by said predetermined programmable machine.

7. A process according to claim 6, wherein said code is a computer program source code.

8. A process as in claim 6 or claim 7, wherein said change causes said fixed block of memory to be addressed from a predetermined one of its two ends depending upon the convention utilized by said predetermined programmable machine for ordering the significance of bytes within words.

9. A process as in any one of claims 6 to 8, wherein said change causes the fixed block of memory contents for a big-endian machine to be inverted to the mirror image of that for a little-endian machine.

10. A method of endian transformation substantially as hereinbefore described.

11. A system for providing endian transformation substantially as hereinbefore described.

12. A process for compiling or translating a complete program code substantially as hereinbefore descried with reference to the accompanying drawings.

INTERNATIONAL SEARCH REPORT

Int: :ional Application No
PCT/GB 99/03167

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F9/34

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Stallings: 'Endian Issues', BYTE, 9/95, downloaded from the Internet on 7.1.2000; URL: www.byte.com/art/9509/sec12/art1.htm (and associated images). XP002127175	1-3,5-9
A	Paragraph: 'Byte Storage'	4
X	EP 0 764 899 A (SIEMENS AG) 26 March 1997 (1997-03-26)	1-3,5-9
A	claim 1	4

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
11 January 2000	28. 01. 2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040. Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Cohen, B

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Int[illegible]ional Application No
PCT/GB 99/03167

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	COHEN N H: "ENDIAN-INDEPENDENT RECORD REPRESENTATION CLAUSES" ADA LETTERS,US,ASSOCIATION FOR COMPUTING MACHINERY, NEW YORK, NY, vol. 14, no. 1, page 27-29 XP000610792 ISSN: 0736-721X	1-3,5-9
A	the whole document	4
A	Motorola: PowerPC 604e, RISC Microprocessor User's Manual; 3/98; pages 1-12 and 1-13. XP002127176 paragraph '1.3.2.2!	1-9

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inte. ..tional application No.
PCT/GB 99/03167

Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☒ Claims Nos.: 10-12
because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:
see FURTHER INFORMATION sheet PCT/ISA/210

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. ☐ As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest.

☐ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

International Application No. PCT/GB 99 /03167

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

Continuation of Box I.2

Claims Nos.: 10-12

Claims 10-12 do not contain any technical features. Consequently, these claims do not define the matter for which protection is sought in terms of the technical features of the invention (Art. 84, Rule 29(1)). This deficiency is to such an extent, that a meaningful search of these claims is impossible.

The applicant's attention is drawn to the fact that claims, or parts of claims, relating to inventions in respect of which no international search report has been established need not be the subject of an international preliminary examination (Rule 66.1(e) PCT). The applicant is advised that the EPO policy when acting as an International Preliminary Examining Authority is normally not to carry out a preliminary examination on matter which has not been searched. This is the case irrespective of whether or not the claims are amended following receipt of the search report or during any Chapter II procedure.

INTERNATIONAL SEARCH REPORT

information on patent family members

Inte ional Application No
PCT/GB 99/03167

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0764899 A	26-03-1997	DE 19535306 A	27-03-1997